

# A learning algorithm for a hybrid nonlinear predictor applied to noisy nonlinear time series

著者	Khalaf Ashraf A.M., Nakayama Kenji
journal or publication title	IEEE&INNS Proc. IJCNN'99, Washington,DC
page range	1590-1593
year	1999-07-01
URL	<a href="http://hdl.handle.net/2297/6845">http://hdl.handle.net/2297/6845</a>

# A Learning Algorithm for A Hybrid Nonlinear Predictor Applied to Noisy Nonlinear Time Series

Ashraf A.M. Khalaf      K. Nakayama

Graduate School of Natural Science and Technology, Kanazawa University,  
Kanazawa-shi, 920-8667, Japan. e-mail: nakayama@t.kanazawa-u.ac.jp

## Abstract

*A hybrid nonlinear time series predictor was proposed, in which a nonlinear sub-predictor (NSP) and a linear sub-predictor (LSP) are combined in a cascade form. In this paper, We propose a separate learning method, in which the NSP is trained until convergence, then the LSP is trained using the final NSP weights. If the NSP and the LSP are trained simultaneously, the input of the LSP will be far from the correct prediction at the early iterations. This causes disturbance in the LSP learning process. The proposed separate learning method gives better results than the simultaneous one. Furthermore, a new learning algorithm for the NSP is proposed. By enforcing the NSP weights and biases to take large values until a certain number of the learning iterations, the input potential of the hidden neurons are expanded and shifted towards the saturation regions of the sigmoid functions. As a result, noise effects can be suppressed. Computer simulations, using real world time series, demonstrates usefulness of the proposals.*

## 1. Introduction

The linear signal processing tools are insufficient to deal with nonlinear time series processing. On the other hand, neural networks are useful for nonlinear adaptive signal processing. They have been applied successfully in a variety of signal and information processing fields [1],[2]. One of these fields is the nonlinear time series prediction [3]-[7].

We have considered the nonlinear prediction problem as a pattern mapping one. A multi-layer neural network, which consists of sigmoidal hidden neurons and a single linear output neuron, has been employed as a nonlinear sub-predictor (NSP). Since the NSP includes nonlinear functions, it can predict the nonlinearity of the input series. However, the prediction is not complete in some cases. So, the prediction error of the NSP is further compensated for by employing a linear sub-predictor (LSP) after the NSP. Also, a nonlinearity analysis method for the time series has been proposed to estimate the predictor size [8]-[10].

In this paper, we propose a learning methodology for the hybrid nonlinear predictor [8]-[10] to obtain good generalization. This method is called a separate learning method, in which the NSP is trained until a certain number of learning iterations, then the LSP is trained using the final NSP weights. A common learning method called a simultaneous learning method, in which the NSP and the LSP are trained simultaneously, is investigated for comparison. Both the proposed separate learning method and the simultaneous

learning method are based on the normal back propagation algorithm.

Furthermore, we propose a new enhancement back-propagation algorithm to be applied in the noisy environment. The purpose of the proposed algorithm is to suppress noise effects on the predictor performance using the nonlinearity of the activation functions used in the hidden neurons of the NSP.

Computer simulations, using real world time series are demonstrated.

## 2. Hybrid Predictor

### 2.1 Network Structure

Figure 1 demonstrates the structure of the hybrid predictor [11]. As a first stage of the predictor, we employ a multi-layer neural network (MLNN), which is good for pattern mapping. It is called a nonlinear sub-predictor (NSP). It consists of sigmoidal hidden neurons and a single linear output neuron. The NSP is trained by the supervised learning algorithm using the sample  $x(n)$  to be predicted as the target. This means the NSP itself is trained as a single predictor.

Since the NSP includes nonlinear functions, it can predict the nonlinearity of the input time series. However, the prediction is not complete in some cases. So, the NSP prediction error is further compensated for by employing a linear finite impulse response (FIR) sub-predictor (LSP) after the NSP. The LSP is trained by using  $x(n)$  as the target too. Thus, the same target is used for both the NSP and the LSP.

The reason why we use  $x(n)$  as the target for the NSP is explained as follows: First, it is difficult to obtain the target only for nonlinear prediction. Second, since the NSP has a linear output neuron, the linear prediction is also possible to some extent. So, nonlinear and some part of linear properties of the input signal can be predicted by the NSP and the remaining part is predicted by the LSP.

### 2.2 System Equations of NSP

The output of the  $j$ th hidden neuron  $v_j(n)$  at time  $n$  is expressed by

$$u_j(n) = \sum_{i=1}^N w_{ji} x(n-i) + \theta_j, \quad (1)$$

$$v_j(n) = f_h(u_j(n)), \quad j = 1, 2, \dots, L, \quad (2)$$

where  $w_{ji}$  is the connection weight from the  $i$ th input node

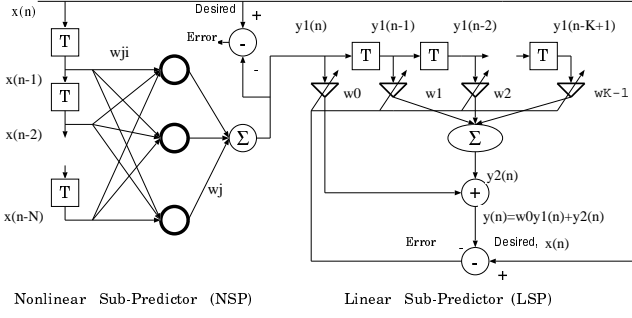


Fig. 1. Structure of the hybrid predictor.

to the  $j$ th hidden neuron.  $L$  is the number of the hidden neurons and  $\theta_j$  is the biases. The activation function  $f_h()$  used in the hidden neurons is a sigmoid function given by

$$f_h(x) = \frac{1}{1 + \exp(-x)}. \quad (3)$$

The output layer contains only one linear neuron. Its output  $y_1(n)$  is expressed by

$$u(n) = \sum_{j=1}^L w_j v_j(n) + \theta, \quad (4)$$

$$y_1(n) = f_o(u(n)) = u(n), \quad (5)$$

$w_j$  is the connection weight from the  $j$ th hidden neuron to the output neuron and  $\theta$  is the biases. The connection weights  $w_{ji}$  and  $w_j$ , and the biases  $\theta_j$  and  $\theta$  in the NSP are adjusted by the back-propagation algorithm.

On the other hand, the LSP coefficients are adjusted by the least mean square (LMS) algorithm.

### 2.3 Prediction Error Evaluation

The prediction error of the NSP is

$$e_{NSP}(n) = x(n) - y_1(n). \quad (6)$$

The instantaneous squared error of the NSP is

$$\xi_{NSP}(n) = \frac{1}{2} e_{NSP}^2(n). \quad (7)$$

The mean square error  $MSE$  over an epoch is

$$MSE_{NSP} = \frac{1}{M} \sum_{n=1}^M \xi_{NSP}(n), \quad (8)$$

where  $M$  is the number of samples in one epoch. The mean squared error at the LSP output is calculated by the same way. The normalized root-mean-square error ( $NRMSE$ ) will be used to express the prediction error.

$$NRMSE = \sqrt{MSE/P_s}, \quad (9)$$

$MSE$  indicates the mean squared error at the output of the NSP and the LSP.  $P_s = (\mathbf{x}^T \mathbf{x})/M$  is the input signal power.  $\mathbf{x}$  is the vector contains the input samples.  $T$  is the transposition operator.

## 3. Learning Methods for Noise-free and Noisy Data

### 3.1 A Separate Learning Method for Noise-free Data

The NSP can predict nonlinearity and some part of linearity of the time series. Therefore,  $x(n)$ , that is the target for the hybrid predictor, is used for the NSP too. In other words, the NSP has an independent target and is trained independently. On the other hand, the LSP is trained so as to compensate for the prediction error caused by the NSP.

In a simultaneous learning method, in which the NSP and LSP are trained at the same time, the LSP input is far from the correct prediction at the beginning. Then it will cause a disturbance in the LSP learning process.

We propose a separate learning method. This learning process consists of the following two stages:

*Stage 1:* The NSP is trained using  $x(n)$  as a target by the normal back-propagation algorithm (NBP).

*Stage 2:* The NSP is fixed as the above result. The LSP is trained using  $x(n)$  as a target by LMS algorithm.

These two learning methods will be compared through computer simulation.

### 3.2 An Enhanced Algorithm for Noisy Data

In measuring physical phenomena, data transmission and processing, noise cannot be avoided. Therefore, in real world applications, noise effects must be investigated.

One learning method to suppress noise effects is to use many noisy data in the learning phase. Another method, in which frequency bands of signal and noise are separated, then the noise can be filtered by linear filtering. Furthermore, if saturation type nonlinear functions are used in the hidden neurons, the noise effects can be suppressed by shifting or expanding the input potential of hidden neurons toward the saturation regions. The first method requires a huge number training data, and a very long time. The second method is rather difficult for nonlinear time series.

In this section, a learning algorithm based on the third idea is proposed. An enhancement back-propagation algorithm (EBP) is proposed to be applied in the noisy environment. If the input potential  $u_j(n)$  of the hidden neurons are located in the saturation regions of the sigmoid function  $f_h()$  in Eq.(3), effects of the noise included in the input time series can be suppressed. Since  $u_j(n)$  is given by Eq.(1), it can be shifted or expanded toward the saturation regions by enlarging  $w_{ji}$  and  $\theta_j$ . For this reason, the following enhanced learning method is proposed for the NSP.

*Step 1:* The NSP is trained by the back-propagation algorithm in an ordinary fashion.

*Step 2:* The NSP trained in Stage 1 is further trained

through the modified back-propagation algorithm, in which the following enhancement is embedded.

Let  $w_{ji}(n)$  and  $\theta_j(n)$  be the connection weights and the biases updated at the  $n$ -th epoch through the back-propagation algorithm. At the same epoch, they are further enhanced as follows:

$$\begin{aligned} (1+r^n)w_{ji}(n), \quad 0 < r < 1, \\ (1+r^n)\theta_j(n), \quad 0 < r < 1. \end{aligned} \quad (10)$$

These values are denoted  $w_{ji}(n)$  and  $\theta_j(n)$ , respectively, once more, and are used in the  $(n+1)$ -th epoch of the back-propagation algorithm.  $r$  is determined by experience, resulting in a small value. This means effect of  $r^n$  will be vanished within some earlier epochs.

*Step 3:* After the NSP learning convergence, the LSP is trained by the LMS algorithm with fixed NSP weights and biases obtained above.

## 4. Simulation Results and Discussions

### 4.1 Noise Effects in Nonlinear Prediction

#### *Training phase:*

It is assumed that we can get noise-free time series and a probability distribution function of noise. The training is carried out using the noisy data as the input and the noise-free data as the target. The noise used here is Gaussian white noise. The training data sets are prepared by adding 10 noise sets to the noise-free time series. So, 10 noisy training data sets are used in one epoch.

Effect of the training using the noisy time series is evaluated. Especially, distribution of the input potential of the hidden neurons, that is,  $u_j(n)$  given by Eq.(1), is investigated. The sigmoidal functions are used in the hidden neurons. If  $u_j(n)$  for all the noisy data can be distributed mainly in the saturation regions, then noise effects can be suppressed.

#### *Testing phase:*

After the training, the predictors are tested using other samples of the time series and the noise, which are not used in the training phase. In this phase, the input signal and the target are the noisy data and the noise-free data, respectively.

Furthermore, we employ the following reference in order to evaluate the prediction performance.

$$R = \sqrt{(MSE_{nf} + P_n)/P_s}. \quad (11)$$

$MSE_{nf}$  and  $P_n$  are the mean square prediction error for the noise-free time series and the noise power, respectively. The meaning of  $R$  is the following: Since

the noise used here is white noise, which cannot be predicted, the noise will remain just as it is. Therefore, the mean squared prediction error becomes  $MSE_{nf} + P_n$ . If the prediction error obtained in the simulation is less than  $R$ , then the noise effect can be compensated for through the training. The predictor becomes robust against the additive noise.

### 4.2 Nonlinear Time Series

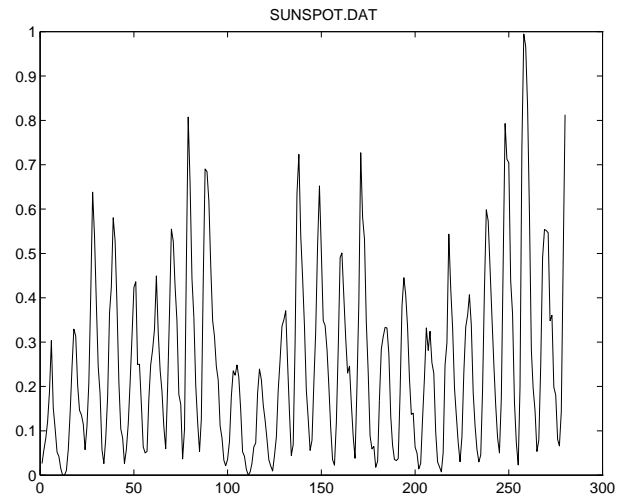


Fig. 2. Sunspot time series from 1700 to 1979

Computer simulations have been done for a one-step forward prediction task for Sunspot data shown in Fig.2. Data file of Sunspot time series was downloaded from Santa Fe public home page. We have used the record of the sunspot data from the year 1700 to 1920 (221 samples) in the training phase and the data from 1921 to 1979 (59 samples) in the testing phase. The same data were used in [3] and others.

### 4.3 Prediction performance

TABLE I  
NRMSE IN TESTING PHASE USING SUNSPOT DATA.  
 $S/N = 29.5dB$

Learning	Testing	NBP-I	NBP-II	EBP
NF+NF	NF+NF	0.2074	0.1684	0.1539
NF+NF	N+NF	0.2088	0.1702	0.1558
N+NF	N+NF	0.2334	0.1864	0.1675

Table I shows the  $NRMSE$  for the hybrid predictor trained by the normal back-propagation in both the si-

multaneous (NBP-I) and the separate (NBP-II) learning procedures and the enhanced back-propagation (EBP) algorithms with  $r=0.01$  in Eq.(10). In this table, for example, N+NF means the noisy and the noise-free data are used for the input and the target, respectively. Sunspot data are used with the signal to noise ratio  $S/N=29.5\text{dB}$ .

We have three points to be clarified. First, in the first row, using the noise-free data for both the input and the target, NBP-II is better than NBP-I. In the second row, using the noisy input data in the testing phase, the  $NRMSE$  is slightly increased. However, NBP-II is still better than NBP-I. The third point, using the noisy input data in both the learning and the testing phases is not useful for noise effect reduction. At the same time the EBP algorithm gives the best prediction performance and noise effect suppression.

TABLE II  
 $NRMSE$  FOR NOISY SUNSPOT DATA USING HYBRID PREDICTOR  
TRAINED BY NORMAL AND ENHANCED BACK-PROPAGATION  
ALGORITHMS.  $S/N = 29.5\text{dB}$ .

Learning	Testing	NBP-II	EBP	$R$
NF+NF	N+NF	0.1702	0.1558	0.1720
N+NF	N+NF	0.1864	0.1675	

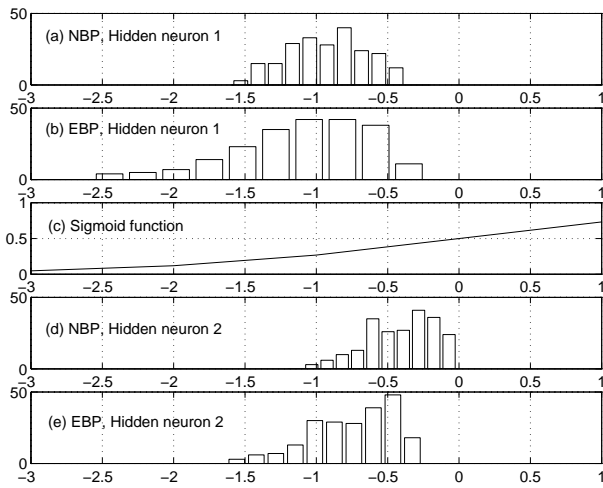


Fig. 3. Histogram of input potential for two hidden neurons of NSP trained by NBP and EBP algorithms. Sunspot noise-free data are used.

Table II compares between  $R$  defined by Eq.(11) and the  $NRMSE$ . Since  $S/N=29.5\text{dB}$ ,  $R$  becomes 0.172. The  $NRMSE$  obtained by the NBP-II using the noise-free training data is almost the same as  $R$ . This means the white noise effects cannot be suppressed. It re-

mains as it is. When the noisy training data are used, the  $NRMSE$  is increased.

On the contrary, the EBP algorithm with  $r=0.01$  in Eq.(10), can reduce the  $NRMSE$  under  $R$ . This means noise effects can be suppressed by the new enhanced algorithm. In this enhanced algorithm, still using the noisy training data is not useful.

Figure 3 shows the histogram of the input potential  $u_j(n)$  of two hidden neurons. The noise-free data are used in the learning phase. The distribution of  $u_j(n)$  obtained by the EBP algorithm are expanded and shifted toward the saturation region of the sigmoidal function compared to those obtained by the NBP-II algorithm.

## 5. Conclusion

In this paper, a separate learning method has been proposed for the hybrid nonlinear predictor. This learning method gives better results than the simultaneous one. Furthermore, a new enhanced back-propagation algorithm has been proposed for noisy data. The training data can be drastically saved, while generalization is guaranteed. The connection weights and the biases are slightly enhanced to reduce the noise effects.

## REFERENCES

- [1] S.Haykin, "Neural Networks: A Comprehensive Foundation," Macmillan, New York, 1994.
- [2] J. Hertz, A. Krogh, and R. G. Palmer, "Introduction to the Theory of Neural Computation," Santa Fe Institute, 1991.
- [3] A. S. Weigend and D. E. Rumelhart, "Generalization through minimal networks with application to forecasting," in Proc. INTERFACE'91 : Computing Science and Statistics, ed. Elaine Keramindas, pp.362-370, Springer Verlag, 1992.
- [4] S.Haykin and L. Li, "Nonlinear adaptive prediction of nonstationary signals," IEEE Trans. Signal Processing, vol.43, No.2, pp.526-535, February, 1995.
- [5] A. S. Weigend and N. A. Gershenfeld, Time series prediction: Forecasting the future and understanding the past. in Proc. V. XV, Santa Fe Institute, 1994.
- [6] T.J. Cholewo and J.M. Zurada, "Sequential network construction for time series prediction," in Proc. ICNN'97, pp.2034-2038, 1997.
- [7] X.M. Gao, X.Z. Gao, J.M.A. Tanskanen, and S.J. Ovaska, "Power prediction in mobile communication systems using an optimal neural-network structure," IEEE Trans. on Neural Networks, vol.8, No.6, pp.1446-1455, November 1997.
- [8] A. A.M.Khalaf, K.Nakayama, and K. Hara, "A Neural-FIR Predictor: Minimum Size Estimation Based on Nonlinearity Analysis of Input Sequence," in Proc. of ICANN'97, Lausanne, Switzerland, pp. 1047-1052, October 1997.
- [9] A.A.M.Khalaf, K.Nakayama, "A Cascade Form Predictor of Neural and FIR Filters and Its Minimum Size Estimation Based on Nonlinearity Analysis of Time Series," IEICE TRANS. Fundamental, vol. E81-A, No. 3, pp. 364-373, March 1998.
- [10] A.A.M.Khalaf, K.Nakayama, "Time Series Prediction Using a Hybrid Model of Neural Network and FIR filter," in Proc. of IJCNN'98, Anchorage, Alaska, pp.1975-1980, May 1998.